

Algoritmos y Estructuras de Datos II

Segundo parcial

Sábado 15 de mayo de 2021

Aclaraciones

- El parcial es individual y a libro abierto.
- Se responderán consultas por Discord, sólo de interpretación del enunciado, a través del canal ‘Mesa de Docentes’ de la categoría PARCIALES.
- Cualquier aclaración sobre el enunciado que se haga se realizará a través del canal ‘anuncios-evaluaciones’.
- El parcial durará 3 horas, de 9 a 12hs, y la entrega se realizará a través del Campus hasta esa hora, en forma estricta. Se desestimarán entregas fuera de tiempo, sin excepciones.
- Cada ejercicio se calificará con **P**erfecto, **A**probado, **R**egular, o **I**nsuficiente.
- Para aprobar la cursada se deben tener todos los ejercicios aprobados de todos los parciales.
- Los ejercicios con **R** o **I** se recuperarán por separado al final del cuatrimestre.

Ej. 1. Complejidad

Ej. 1.1. Funciones

Sean k y c dos constantes enteras mayores a 1.

Marque **todos** los items correctos. Demuestre por qué los items marcados son correctos y por qué los otros no lo son.

- $n^k \in O(c^n)$
- $n^k \in \Omega(c^n)$
- $n^k \in \Theta(c^n)$

- $\log_k n \in O(\log_c n)$
- $\log_k n \in \Omega(\log_c n)$
- $\log_k n \in \Theta(\log_c n)$

- Si $k > c$ entonces $(cn)^k \in O((kn)^c)$
- Si $k > c$ entonces $(cn)^k \in \Omega((kn)^c)$
- Si $k > c$ entonces $(cn)^k \in \Theta((kn)^c)$

- Si $k > c$ entonces $c^n \in O(k^n)$
- Si $k > c$ entonces $c^n \in \Omega(k^n)$
- Si $k > c$ entonces $c^n \in \Theta(k^n)$

Ej. 1.2. Matrices en degradé

Se tiene una matriz *en degradé*, por ejemplo, llamada A , de $n \times n$ números naturales, de manera que $A[i, j]$ representa al elemento en la fila i y columna j ($1 \leq i, j \leq n$). Se sabe que el acceso a un elemento cualquiera se realiza en tiempo $O(1)$, así como la obtención de la dimensión de la matriz (función LONG). Una matriz *en degradé* es una en que todos los elementos de la matriz son distintos y que todas las filas y columnas de la matriz están ordenadas de forma creciente (es decir, $i < n \Rightarrow A[i, j] < A[i + 1, j]$ y $j < n \Rightarrow A[i, j] < A[i, j + 1]$), como se aprecia en los ejemplos A_1 y A_2 de más abajo.

2	14	70	318	3464
5	41	159	839	7269
53	239	596	1514	21901
151	1114	2969	8878	79094
412	4431	8971	35720	134696

Cuadro 1: Una matriz de ejemplo (A_1)

1	4	35	157
5	19	118	334
9	64	464	1395
54	169	1295	5698

Cuadro 2: Otra matriz de ejemplo (A_2)

función VALORENMATRIZ(matriz de enteros A , entero val)

```

     $n := \text{LONG}(A)$ 
     $i := 0$ 
    mientras  $i < n$  y  $A[0, i] = < v$  hacer
        |  $i := i + 1$ 
    fin
     $colLim := i - 1$ 
     $i := 0$ 
    mientras  $i < n$  y  $A[i, 0] = < v$  hacer
        |  $i := i + 1$ 
    fin
     $filLim := i - 1$ 
    para  $i := 0 \dots filLim$  hacer
        | para  $j := 0 \dots colLim$  hacer
            | | si  $A[i, j] = val$  entonces
                | | | devolver true
            | | fin
        | fin
    fin
    devolver false
fin

```

Realizar el análisis **completo**¹ para los siguientes casos de VALORENMATRIZ. Acotar la complejidad temporal de la manera más ajustada posible. Justificar.

1. Mejor caso
2. Peor caso

¹usando las notaciones vistas en la materia

Ej. 2. Invariante de representación y función de abstracción

Contamos con el siguiente TAD que modela la red social Bookface, que es una alternativa a otra que tiene nombre parecido, pero orientada para un público más joven. En Bookface, cualquiera se puede registrar libremente y no hace falta tener una cuenta de correo para hacerlo. Cualquier usuario le puede enviar una solicitud de amistad a otro usuario (pero no a sí mismo). Si el receptor de la solicitud la acepta, ambos usuarios pasan a ser amigos. Mientras no lo haga, el emisor puede seguir mandando solicitudes de amistad. Un usuario también puede seguir a otro usuario (de nuevo, no puede seguirse a sí mismo). A diferencia de la relación de amistad, la relación de seguimiento no necesariamente es simétrica y no requiere confirmación. Notar que las relaciones de amistad y de seguimiento son independientes. Es decir, puede darse que un par de amigos se sigan entre sí, uno siga al otro pero no al revés, o ninguna de ellas, así como puede pasar que no haya relación de amistad entre un seguidor y un seguido.

TAD Usuario es String

TAD BOOKFACE

observadores básicos

usuarios : bookface \rightarrow conj(usuario)
 amigos? : bookface $b \times$ usuario $u \times$ usuario $v \rightarrow$ bool $\{u, v \in \text{usuarios}(g)\}$
 solicitudesDeAmistadPendientes : bookface $b \times$ usuario $u \rightarrow$ multiconj(usuario) $\{u \in \text{usuarios}(g)\}$
 seguidores : bookface $b \times$ usuario $u \rightarrow$ conj(usuario) $\{u \in \text{usuarios}(g)\}$

generadores

nuevaRed : \rightarrow bookface
 registrarUsuario : bookface $b \times$ usuario $v \rightarrow$ bookface $\{\neg v \in \text{usuarios}(b)\}$
 solicitarAmistad : bookface $b \times$ usuario $u \times$ usuario $v \rightarrow$ bookface $\{u \neq v \wedge u, v \in \text{usuarios}(b) \wedge_L \neg \text{amigos?}(b, u, v)\}$
 aceptarAmistad : bookface $b \times$ usuario $u \times$ usuario $v \rightarrow$ bookface $\{u \neq v \wedge u, v \in \text{usuarios}(b) \wedge_L v \in \text{solicitudesDeAmistadPendientes}(b, u)\}$
 seguir : bookface $b \times$ usuario $u \times$ usuario $v \rightarrow$ bookface $\{u, v \in \text{usuarios}(b) \wedge_L \neg v \in \text{seguidores}(b, u)\}$

...

Fin TAD

Queremos representar la red social mediante la siguiente estructura. En ella, representamos las amistades con tuplas de usuarios (no importa el orden pero no debería haber repetidas). Además, usamos diccionarios indexados por usuario para que, por cada uno, podamos conocer su conjunto de seguidores, las solicitudes de amistad que tiene pendientes (representando cada una con el usuario emisor de dicha solicitud) y la cantidad de amigos que tiene.

bookface **se representa con** estr, donde

estr es tupla(
 amistades : conj(tupla(usuario, usuario)),
 seguidores : dicc(usuario, conj(usuario)),
 solicitudesPendientes : dicc(usuario, secu(usuario)),
 cantAmigos: dicc(usuario, nat)
)

1. Escribir en forma coloquial el invariante de representación.
2. Escribir formalmente el invariante de representación.
3. Escribir formalmente la función de abstracción.